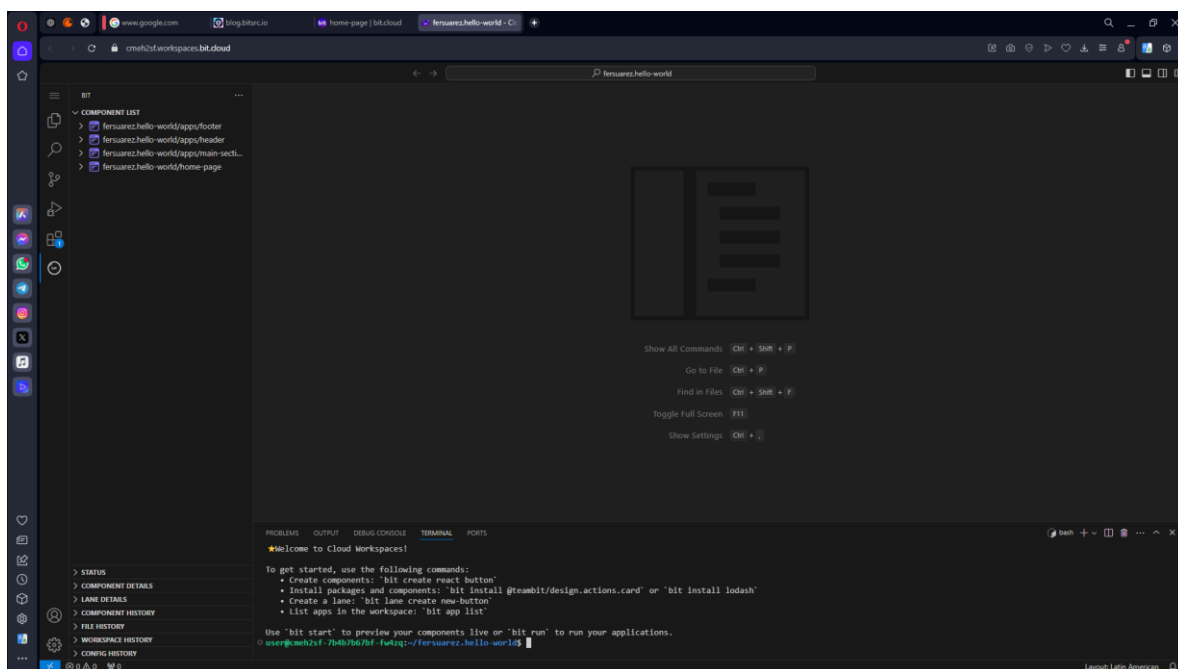


Taller Microfrontends: Bit

Bienvenidos al taller de Microfrontends en BIT. En esta sesión, nos enfocaremos en aprender los conceptos básicos de los microfrontends y su aplicación práctica. El objetivo será implementar un simple "Hola Mundo" utilizando esta arquitectura, lo que permitirá entender cómo fragmentar una aplicación en componentes independientes que pueden desarrollarse y desplegarse por separado.

Introducción a BIT

1. Para iniciar, debes crear una cuenta en [Bit Cloud](#) para utilizar todas las herramientas que proporciona esta plataforma (Workspace, scopes, graphs, visualizaciones)
2. Una vez creada la cuenta, deberás crear un Scope en donde abarcarás la aplicación y sus diferentes componentes
3. Ahora, deberás crear un Workspace y se verá así cuando lo configures:



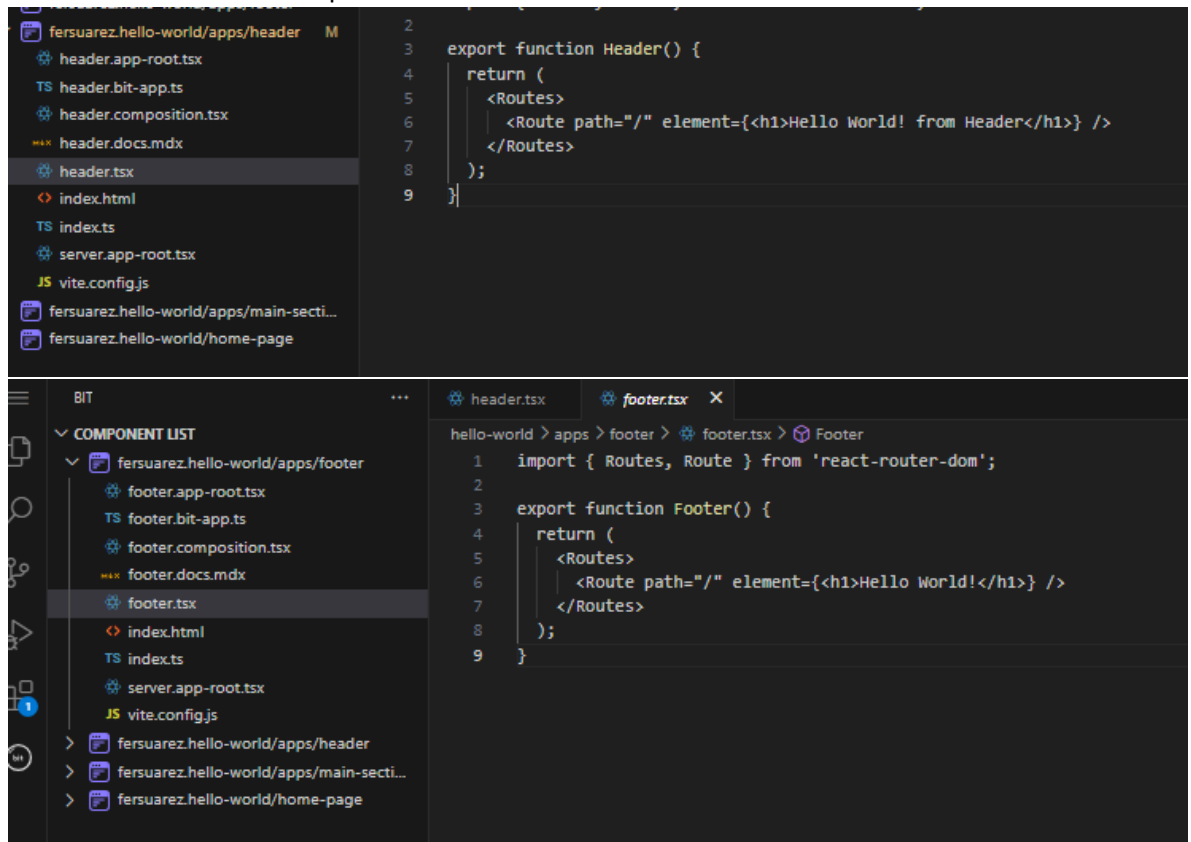
Este será nuestro entorno de trabajo par la creación de los diferentes componentes

Creación de parcels

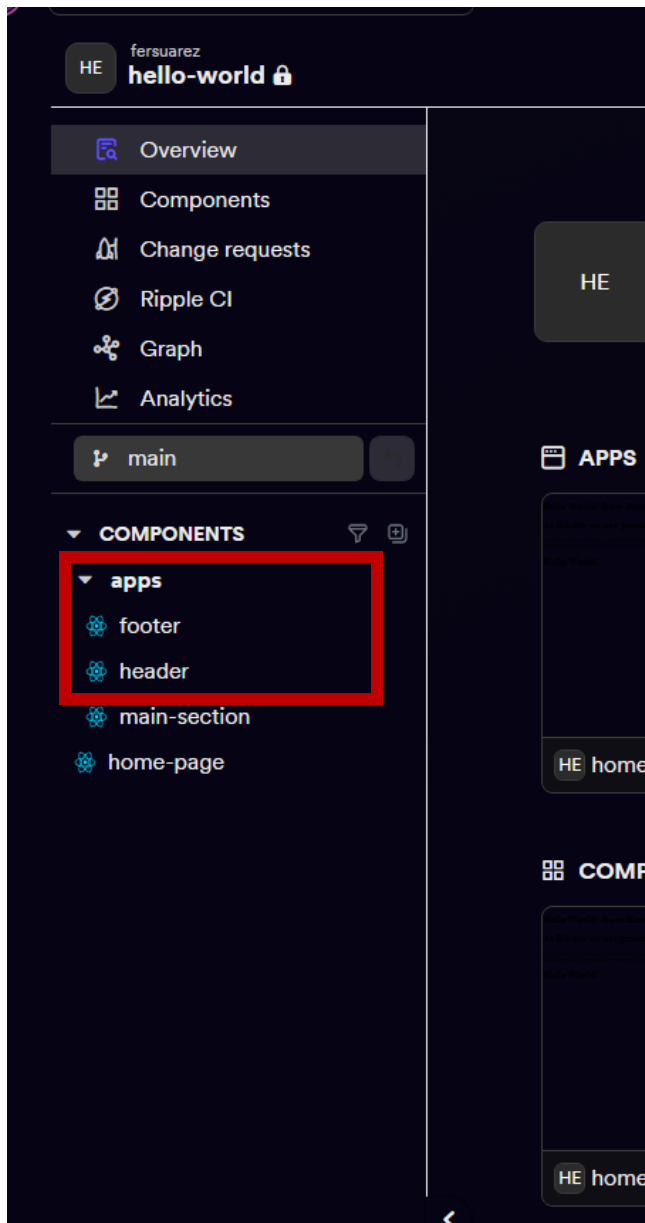
Iniciaremos con los componentes Footer y Header, en donde ambos deberán ser creados siguiendo los siguientes puntos desde la terminal del workspace (VS code) de BIT:

1. Creamos la parcel con el comando: ***bit create react-app apps/name_component***
También se puede con otros frameworks pero este taller estará enfocado en React

2. Antes de comenzar a escribir el código para el footer o header, deberás hacer algo rápidamente para que tu espacio de trabajo pueda cargar el componente ejecutando este comando. ***bit use username.name_scope /apps/name_component***
3. Para corroborar que se crearon bien los componentes ejecutarás el comando de ***bit app list***
4. Luego haremos un cambio pequeño en el archivo de ***name_component.tsx*** para “diferenciar” nuestros componentes:



5. Realizamos el build de cada componente para dejarlo listo para exportar con estos últimos cambios con el comando de ***bit build use username.name_scope /apps/name_component***
6. Como siguiente paso, le asignaremos un tag al componente con el siguiente comando: ***bit tag username.name_scope /apps/name_component***
7. Y para finalizar, haremos un ***bit export*** para exportar todos los componentes que tengamos en uso.
8. Después de unos segundos, en la plataforma de BIT se verá algo así en el panel lateral:

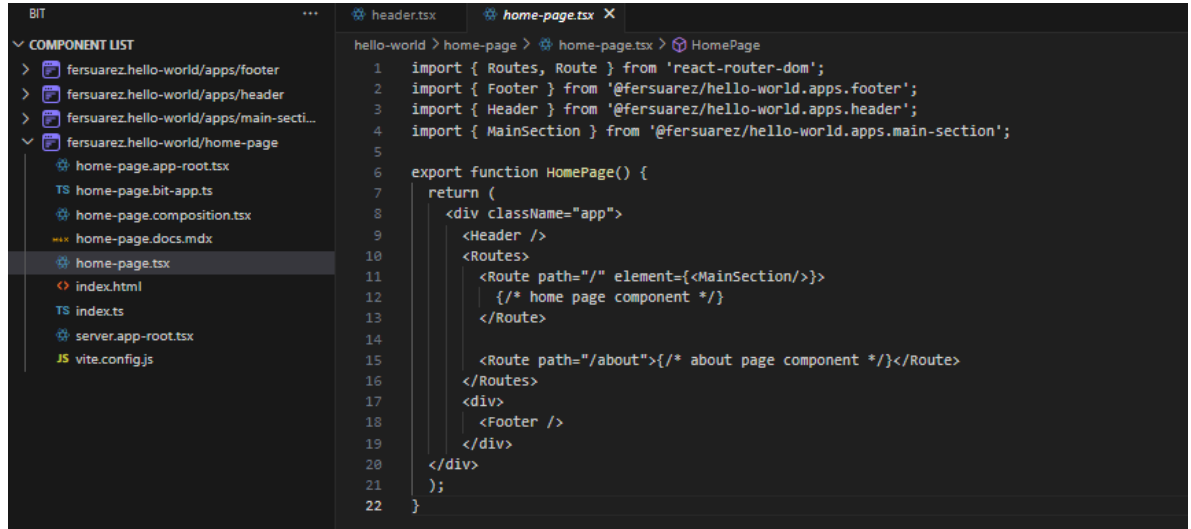


Creación de Shell:

Una vez tienes los componentes creados y publicados correctamente, ya podrás importarlos en cualquier app que esté dentro de tu scope. Para ello, deberás seguir los siguientes pasos:

1. Creas el Shell con el comando ***bit create react-app home-page***
2. Al igual que las parcels, debes indicarle al entorno de trabajo que vas a usar dicho artefacto con ***bit use username.name_scope /home-page***

3. Tendrás que modificar el archivo de `home-page.tsx` para importar las parcels desarrolladas anteriormente de la siguiente manera:

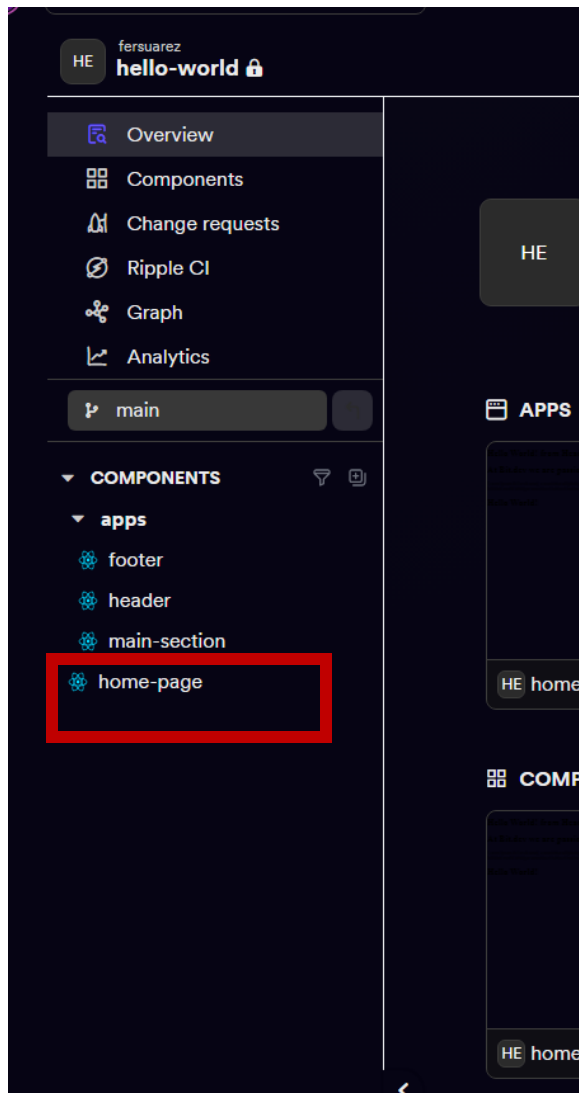


```
BIT *** header.tsx home-page.tsx X
COMPONENT LIST
> fersuarez.hello-world/apps/footer
> fersuarez.hello-world/apps/header
> fersuarez.hello-world/apps/main-secti...
▼ fersuarez.hello-world/home-page
  home-page.app-root.tsx
  TS home-page.bit-app.ts
  home-page.composition.tsx
  *** home-page.docs.mdx
  home-page.tsx
  index.html
  TS index.ts
  server.app-root.tsx
  JS vite.config.js

hello-world > home-page > home-page.tsx > HomePage
1 import { Routes, Route } from 'react-router-dom';
2 import { Footer } from '@fersuarez/hello-world.apps.footer';
3 import { Header } from '@fersuarez/hello-world.apps.header';
4 import { MainSection } from '@fersuarez/hello-world.apps.main-section';
5
6 export function HomePage() {
7   return (
8     <div className="app">
9       <Header />
10      <Routes>
11        <Route path="/" element={<MainSection/>}
12          { /* home page component */ }
13        </Route>
14
15        <Route path="/about">{ /* about page component */}</Route>
16      </Routes>
17      <div>
18        <Footer />
19      </div>
20    </div>
21  );
22 }
```

Donde cada componente aparecerá con su respectiva etiqueta o alias que le asignes

4. Allí, lo único que hace falta es realizar nuevamente el build y el taggeo de nuestro componente con los comandos de **`bit build username.name_scope/home-page`** y **`bit tag username.name_scope/home-page`** respectivamente
5. Y como último paso, ya solo te queda realizar el **`bit export`** para ya publicar tu contenedor principal
6. Al igual que con las parcels, toma un tiempo en verse reflejado el cambio en nuestro entorno de bit.cloud, pero una vez está listo, aparece algo así en nuestro panel lateral:



Con esto, estarías terminando la creación de y culminación del taller. Adentro de dicha plataforma, hay varias herramientas para la visualización (por componentes o global), visualización de un gráfico donde se muestra el árbol de aplicaciones creadas y muchas más herramientas... nos gustaría recibir comentarios sobre cuáles son esas herramientas que más te impactaron, ventajas y desventajas de BIT o cualquier comentario respecto al taller. No siendo más, te deseamos un ...

